Spring 2008

CS 400/600: Data Structures and Algorithms

Eric Maston
Wright State University - Main Campus

Computer Science 400/600
Spring 2008
Wright State University

March 31, 2008

Course Description

This is the initial course in the understanding of data structures. This course focuses on data structures with abstract data types, such as trees, stacks, queues and graphs. The course will use projects to create a deeper understanding of data structure and their use.

Goals

There are several goals in CS 400/600:
1. Develop deep understanding of data structures.
2. Learn how and when to use data structures in C++.
3. Understand theoretical analysis of algorithms.
4. Learn how to develop complex software programs.
5. Have some fun!

Class Details

Lecturer: Eric Matson
Office: 336 Russ Engineering Center
Phone: 937-775-5108
Office Hours: Monday 9:30 - 10:30, Wednesday 12:30 - 1:30 at Russ 336 or by appt.
Email: eric.matson@wright.edu
Web: WebCT
Class: 2:45 - 4:00 (pm), Russ Engineer Cntr 150
Text: Data Structures and Algorithms in C++, Goodrich, Tamassia and Mount
Platform: Bloodshed Dev C++ Grader: Bhanu Potluri *potluri.6@wright.edu*

Prerequisites

For this class, the official prerequisite is CS 242. Please let me know the first lecture if you do not meet this prerequisite, and we can talk about your preparation if it differs. This section of CS 400 will be taught using the C++ programming language.

Grading

Homework Assignments 20%
Programming Projects 40%
Midterm Exam 20%
Final Exam 20%

The base scale is: A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59. This is the highest requirement that will be used. The scales may be lowered or revised if necessary.

Schedule

(always subject to changes) Always have readings scheduled for that day complete prior to the class meeting

#	Date	Topic	Reading
1	Mar 31	Introduction	1, 2
2	Apr 2	ADT, Vectors, Lists	5.1 - 5.2
3	Apr 7	Stacks and Queues	4.1 - 4.6
4	Apr 9	Algorithm Analysis	3, Appendix A
	Apr 14	Algorithm Analysis - Search	
5	Apr 16	Algorithm Analysis - Sort	
6	Apr 21	General Trees	6.1 - 6.2
7	Apr 23	Binary Trees	6.3 - 6.4
8	Apr 28	Binary Search Trees	9.1
9	Apr 30	Priority Queues and Heaps	7.1 - 7.3
10	May 5	**Midterm Examination**	
11	May 7	More Trees	9.2 - 9.5
12	May 12	Trees	
13	May 14	Graphs	12
14	May 19	Graphs	
15	May 21	Graphs	
16	May 26	Graphs	
17	May 28	Search	
18	Jun 2	Search	
19	Jun 4	Hashing	
20	Jun 11	**Final Exam**	3:15 - 5:15 pm

Policies and Notes

- Attendance: Attendance is not required, nor will it be taken after the first couple of lectures. If you are not a regular attendee, it will be your responsibility to seek out what material was covered in the lecture and learn it. Most of my exam questions will be taken directly from ideas covered during the lecture, so it greatly helps if you attend!
- I will utilize WebCT to post updates to the course, sample code, projects, announcements, schedule, etc. Get in the habit of checking it regularly.
- The prerequisites of the course are basic understanding of high-level development in C++ and object oriented concepts. If you are not confident in your skills or do not have the required prerequisites, then visit with me and I can evaluate how to catch your skills up the appropriate level and develop a plan to do so.
- **Always make back ups of all of you work. Never have just one copy of anything!** This way, when your dog eats your laptop the day before the final project is due, you will not have a problem (except getting a new laptop).
- If you are going to miss an exam, for any reason, discuss it with me in advance. If it is an emergency situation, please notify me as soon as possible.
- You can reach me a number of ways. Email is normally the best as I check it about 18 hours a day normally. You can also reach me by phone during the day at 775-5108. If you need human contact either stop in during my office hours, make an appointment, or just come by my office. If I am in and not on a deadline to get something else completed, I will normally try to help as much as possible.
- There are technologies we will use in this class that you may not already know, such as file transfer, command line, text editors, file systems, etc. We will cover some of these technologies as we go.
- The key to learning in this class will be spending time working through the problems. Do not wait until 2 hours before something is due to try to learn the concept and then write the program. This normally ends in a disaster! Stay up with the readings and try to work through some of the examples in the book.

Academic Misconduct

In this class, the only way to truly learn the concepts to is do the work yourself. I encourage working with other people on the course concepts. When you begin to write the program, complete and submit your own work.

Work that has obviously been copied or in the more extreme case, when the original authors name has not even been changed, both parties will receive a 0 grade for that assignment. Both parties will also be turned over to the Office of Judicial Affairs.